

SophiaBeads Datasets Project Documentation and Tutorials

Coban, Sophia Bethany

2015

MIMS EPrint: 2015.26

Manchester Institute for Mathematical Sciences School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/ And by contacting: The MIMS Secretary School of Mathematics The University of Manchester Manchester, M13 9PL, UK

ISSN 1749-9097

SophiaBeads Datasets Project Documentation and Tutorials

Sophia Bethany Coban School of Mathematics, University of Manchester e-mail: sophia.coban@manchester.ac.uk website: www.maths.manchester.ac.uk/~scoban

April 1, 2015

The SophiaBeads Datasets [4] are real microCT datasets, acquired specifically for implementing, testing and comparing iterative reconstruction algorithms. The main motivations for the SophiaBeads Datasets Project are providing real datasets for researchers, and introducing a framework for designing experiments and choosing appropriate reconstruction methods via fair comparisons. This aspect of our work is studied in great detail in [5, 6]. The reason we use the SophiaBeads Datasets is because we know what the reconstructions should look like: We know the insides of the sample and its characteristics, so we can quantify the reconstructions and find out how close we are to an 'exact solution'. The details of the sample and the experiment plan are listed in the next section.

As part of this project, we have also released SophiaBeads Datasets project codes [3]. Our aim with this report is to provide the reader with enough information to work with these codes so the reader can reconstruct the datasets. Additionally, we include a tutorial for quantifying the reconstructions so the readers are able to reproduce our results presented in [5, 6]. In §3, we document the project codes and explain the main script (sophiaBeads.m), followed by documentation for the quantification stage using Avizo. §5.1 and §5.2 are the detailed, step-by-step tutorials for the project codes and the Avizo work, where we reproduce our results for SOPHIABEADS_512_AVERAGED as an example. For accessing the contents of this project (datasets and the source codes) and guidance on referencing, we refer the reader to §2. It is also recommended to read the additional notes and the licenses under which the project is distributed, which are explained in §6.

1 Sample Information and Experiment Plan

For the SophiaBeads experiments, we have set up a basic sample with specific measurements so we are able to quantify the reconstructions. The sample is a plastic tube with a diameter of 25mm, filled with uniform soda-lime glass ($SiO_2 - Na_2O$) beads of diameters 2.5 mm (with standard deviation 0.1 mm). The sample has been scanned in the same conditions for each dataset, where the number of projections is halved after each batch scan (starting from 2048, down to 64 projections, see Table 1 for more information). This follows the set up explained

in [5, 6]. This design allows us to understand the effects of fewer projections, and develop algorithms that deliver quality results when the information content is low (e.g. patient scans with lower dose or rapid data acquisitions for 3D+time experiments).

Number of ProjectionsNumber of Frames (scans per batch)		Acquisition Time (per frame)	Size of Dataset	
2048	1	24 mins and 10 secs	$15.7~\mathrm{GB}$	
1024	2	12 mins and 5 secs	7.8 GB	
512	4	6 mins and 3 secs	$3.9~\mathrm{GB}$	
256	8	3 mins and 2 secs	1.9 GB	
128	16	1 min and 30 secs	$953.6 \mathrm{MB}$	
64	32	$45 \mathrm{secs}$	474.9 MB	

Table 1: The experiment plan and information about the SophiaBeads Datasets.

The datasets are acquired using the Nikon Custom Bay X-ray CT machine located in the Manchester X-ray Imaging Facility. The reader can visit [8] to find out more about this equipment.

2 Download and Referencing Guide

The SophiaBeads Datasets are accessible via Zenodo, which is an open digital repository aimed at preserving and sharing academic/scientific results (these mainly include datasets or software used in a particular study, or supporting texts for a thesis). The reader can download each dataset as a zipped folder from [4]. When extracting the zipped folders, make a note of the path the folder is extracted to (this becomes important later).

Zenodo provides exportation of various citation formats. To cite the datasets, the reader may export the BibTex record by clicking on the link on the right hand side of the SophiaBeads Datasets Zenodo page. It is not necessary to refer to each dataset separately; citing (using the DOI) for any of the datasets *once* is sufficient.

The SophiaBeads Datasets project codes are released on GitHub [9] and published on Zenodo [3]. The project codes can be downloaded by visiting either the GitHub repository page (download as zip or tar.gz files), or the Zenodo page (available only as a zip file). By default, Zenodo takes an archive of the SophiaBeads GitHub repository every time there is a new release. However, we recommend direct downloads from the GitHub repository for the most recent changes applied in-between major releases. The reader will still need to export the BibTex record from the Zenodo page (to get the complete citation including the correct DOI).

The reader will need to download the source project codes in order to work with SophiaBeads Datasets. However, we ask that the reader cites both the datasets and the project codes *separately*, using the relevant DOI.

3 SophiaBeads Datasets Project Codes

As we mentioned earlier, to be able to work with the SophiaBeads Datasets, we have also released project codes for the pre-reconstruction stage. In this section, we introduce these codes, and give detailed descriptions of each script or function¹. The main script in particular, named sophiaBeads.m, contains the relevant functions for all the stages up to (and excluding) the quantification. A visual description of this script is given in the figure below, followed by a list of all scripts and functions, and their definitions.



Figure 1: This is the process tree for the script sophiaBeads.m.

¹In this report, we assume the reader has some experience using MATLAB, including knowing the difference between a regular script and a function. Nevertheless, we adapt a certain behaviour in the report to distinguish between the two: If the name of a script is mentioned, the name will always be followed by the .m extension, e.g. scriptname.m; whereas a function will just be functionname.

Script/Function	Description
• Main	
sophiaBeads.m	A script with all the commands to carry out the reconstruc- tions. We will be going through this script step by step in the tutorial, in §5.1.
Pre-reconstruction	
pre_recon	This function is a collection of commands that return the outputs data (the dataset stored in MATLAB) and geom (the cone beam geometry parameters required for forward and back projectors).
load_nikon	Loads the data taken by Nikon XTek CT machine.
cutDown_data	Cuts the dataset down from $2000 \times 2000 \times 2000$ to $1564 \times 1564 \times 2000$. This is the size of the images during the data acquisition for all the SophiaBeads experiments.
centre_geom and find_centre.m	Functions to find the centre of rotation and apply this as a correction to data and geom before the reconstruction stage. See [7] for the correction algorithm implemented.
• C codes	
CBproject_c.c CBbackproject_c.c project_single.c backproject_single.c jacobs_rays.h	These are essentially the forward and back projectors provided by our colleagues W. Thompson [12] and D. Szotten [11]. These codes adapt Jacob's ray tracing algorithm as explained in [10].
• Interface for the mex files	
CBproject, CBbackproject	Interfaces written to connect a reconstruction script with the forward and back projectors.
• mex files	
CBproject_c.mexa64 CBbackproject_c.mexa64 CBproject_c.mexmaci64 CBbackproject_c.mexmaci64 CBproject_c.mexw64 CBbackproject_c.mexw64	These are the outputs of setup.m, and are used for calculating Ax (forward projector) or A^Tb (back projector).

·	
• remplates	
setup.m	This script outputs the mex files for forward and back pro- jectors. This and the next function are provided to us by our colleague N. Wadeson [13].
cgls_XTek	An example reconstruction script included as a template for the reader. The CGLS algorithm is given in [2].
• Reading/Writing	

read_vol.m	Writing and reading reconstructed volumes. We
write_vol.m	include $\mathtt{write_tiff.m}$ as an extra file, which we do not
write_tiff.m	use in the tutorial.

The reader is reminded that using the doc or the help command in MATLAB for a particular script/function will output more details. For example:

>> help write_tiff
will output
write_tiff
Function to write the reconstructed volume as a set of tiff images.
Each tiff image is a slice in the z-direction.
INPUT:
 vol: Reconstructed volume.
 pathname: Name of the folder where the volume is to be stored.
 filename: Name of file to store the volume as.

experiment_name: Name of the experiment for reconstructing this volume. This is to help distinguish between volumes, avoids overwriting. NOTE: This can be an emptry string. voxels: Size of the volume. type: 'uint8' or 'uint16'. NOTE: This has to be a string.

DEFAULT VALUES:

experiment_name = ''; (empty string)
voxels = [1564 1564 2000]; (2000 slices)
type = 'uint16';

OUTPUT:

foldername: Name of folder the tiff files are saved in.

Copyright (c) 2015 Sophia Bethany Coban Code is available via the SophiaBeads Datasets project. University of Manchester. This is useful for seeing the inputs required by the function and the default values set in the codes.

As mentioned earlier, we have collected all the commands in one script for an easy and quick run (sophiaBeads.m) but the readers are encouraged to experiment by running these functions individually.

4 The Quantification Stage

The quantification of the reconstructed results was done using a commercial software, named Avizo. This application is developed by FEI Visualization Sciences Group, and is popularly used in materials science for interactive data visualization and analysis. Avizo is convenient for working on the SophiaBeads Datasets as it includes a list of measures we can apply to quantify our results. Here, we give a list of actions to be used in the tutorial and their descriptions; and a list of measures that can be used to quantify our results. The results presented in [5] are obtained using Avizo Fire 8, but the tutorial is valid for Avizo Fire 7 and Avizo 9. The workflow we used is outlined in Figure 2. In the quantification tutorial, we assume the reader has no experience using Avizo.



Figure 2: This is the Avizo workflow used in the quantification tutorial. This particular version is Avizo 9. However, the tutorial in §5.2 is valid for Avizo Fire 7 and 8.

The quantification tutorial is essentially the Watershed Tutorial given in [1, pages 271–293],

modified specifically for the SophiaBeads Datasets. We note here that we do not apply any noise reduction or 'filling in the holes' during the segmentation stages, as this would influence the analysis of the reconstructed volume. Below is the list of actions we apply, and their descriptions.

Action	Description
Interactive Thresholding	Creates a binary image using a defined thresholding range of intensity values. The thresholding range we used for all the SophiaBeads reconstructed volumes is (0.11,0.35). This was enough to separate the exterior and the beads, even for SOPHIABEADS_64_AVERAGED.
Chamfer Distance Map	Distance maps are applied to binary images (.thresholded). They output grey level images where each intensity value represents the minimal distance from the object boundary, in voxels. We use the Chamfer Distance Map, which is a discrete map. This cuts down the computation time whilst giving reliable results for this type of sample.
H-Maxima	Takes the distance map (.distmap) as an input, calculates the regional maxima, and outputs the most inner regions within the objects. This is necessary for labeling individual objects in the image.
Labeling	This is used to index all of the disconnected/segmented objects within the image. We repeat this action twice during the quantification stage: First time is to distinguish between the exterior and the objects of interest (our reconstructed volume). Second is to label the separated objects (as individual beads).
NOT	Inverts a grey level image (in our case, the input is .distmap).
Marker-Based Wa- tershed	Watershed is the main tool used in the quantification tutorial. The al- gorithm automatically separates the beads, which are then labeled as in- dividual objects. We use the <i>Marker-Based Watershed</i> algorithm in the tutorials.
AND NOT	This takes two inputs: .thresholded and .watershed, and outputs an image where the separation lines are subtracted from the binary image.
Filter by Measure	Measures individual objects and ranks them according to the chosen mea- sure (see below for the list of measures we believe are suitable for the datasets).
Generate Surface	Generates a 3D surface from a label image (the output for this is .surf).

Surface View		Allows us to visualize the .surf output on the screen.
Ortho Slice		Views the current image as a slice (the horizontal, xy , slice by default, but the user can change this to view vertical slices, xz or yz).
Volumized dering	Ren-	Renders a 3D visualization of the selected volume.

Below is the list of available measures in Avizo Fire, which we can use to quantify the quality of our reconstructed results.

Measure	Description	SophiaBeads Expected Result	
Shape_VA3D	Outputs a shape factor depending on how spherical the object is. For a perfect sphere, the answer is 1.	SophiaBeads are not all perfect spheres (there is a fraction of the beads that are egg-shaped, which would influence the outcome) so we ac- cept a range between 1 to 1.5 as the "perfect" spherical solution.	
EqDiameter	Inspects the diametres of the spherical objects.	This should output the diametres of the beads, which (in the perfect case) would be 2.5 mm.	
Volume3D	Calculates the spherical volume the objects; uses $4\pi r^3/3$ to calculate the volume of a sphere.	We expect the answer for a "perfect" reconstruction of a sphere to be 8.18 mm ³ (or $8.18 \times 10^9 \mu m^3$), with a standard deviation of 1.02 mm (or $1.02 \times 10^9 \mu m^3$).	
Area3D	Outputs the surface area of a 3D object.	The surface area of a "perfect bead" in Sophi- aBeads experiments should be 19.63 mm ² (or $1.9 \times 10^7 \mu m^2$), with a standard deviation of 3.27 mm (or $0.33 \times 10^7 \mu m$).	
Symmetry	Outputs a factor on how symmetric the objects are.	This should be 1 as we expect the recon- structed beads to be symmetric (egg-shaped beads would also influence this analysis but not by a huge factor, similar to SHAPE_VA3D).	
Eccentricity	Similar to the measure above, outputs a factor on how eccentric the shapes of the objects are.	A very low number would mean the shape is not at all eccentric, which is what we would expect for the beads.	

There are more ideas listed for quantifying the SophiaBeads datasets in [5, 6], which are not available as measures in Avizo. The readers are welcomed to use these but are also invited to consider new ideas for quantifying own reconstructions.

5 Reconstruction and Quantification Tutorials

This section aids the readers through the process of reproducing our results, as presented in [5], and more importantly helps the reader understand the exact steps taken in our codes. In this section, we will be repeating our reconstruction experiments for the SOPHI-ABEADS_512_AVERAGED dataset, and present the messages, results, and the Avizo work involved.

Previously, we have mentioned a main script, where all the relevant commands are used to prepare the dataset for reconstructing and then saving in a format appropriate for Avizo (see Figure 1 for the process tree of sophiaBeads.m). We will now go through each command, but also print the outputs of sophiaBeads.m.

5.1 Reconstruction Using the Project Codes

The scripts are written in a way that everything up to the Avizo stage are automated. However, there are some variables the user will have to manually enter in **sophiaBeads.m** before running the script. These are:

- pathname: Declare the path to folder where the dataset is saved.
- filename: Declare the name of the dataset. For these runs, this is declared as SOPHI-ABEADS_512_AVERAGED.
- geom_type: Declare whether to perform a 2D or a 3D reconstruction. Please note that the variable is case-sensitive, and will not work if, e.g., the variable is declared as geom_type = '2d'.
- experiment_name: This is for naming purposes only. The reconstructed volume will take the name <filename>_<experiment_name>.vol. This variable will help distinguish between the reconstruction experiments, and avoid overwriting a previous result.
- slices: This variable is only valid if geom_type = '3D', and will be ignored otherwise. This is to allow the reader to work on a certain number of slices instead of reconstructing the full volume (2000 slices), essentially cutting down the computation time. For our results and quantification in [5], 200 slices were adequate.
- iterations: This is an optional variable, and only used in the code we have written for CGLS. When implementing own reconstruction method, reader is free to replace/remove this variable as they see fit.

For this particular run, the variables above are declared as the following:

```
pathname = '/media/SophiaBeads_Datasets/SophiaBeads_512_averaged/';
filename = 'SophiaBeads_512_averaged';
```

```
geom_type = '3D';
experiment_name = 'CGLS_200slices';
slices = 200;
iterations = 12;
```

Running sophiaBeads.m will trigger the process outlined in Figure 1. We will now go through this by explaining the commands step by step. The first on our list is setup. Note that this is commented out and is only intended to be a template for those wishing to test projector/back projector techniques. If this line is uncommented, the program will output the following:

```
setup;
>> Creating the folder mex/...
>> Building with 'gcc'.
>> MEX completed successfully.
>> Building with 'gcc'.
>> MEX completed successfully.
```

Take caution as this will overwrite the existing projector and back projector mex files. Please read §6 for more information.

Pre-Reconstruction

Next is a function that runs scripts only for the pre-reconstruction phase. This is where the data gets loaded onto MATLAB, the cone beam geometry is set up, and relevant corrections are applied. Below is the list of functions in pre_recon.m, followed by how the function is called within sophiaBeads.m and their outputs in the command window. Please refer to §3 for the descriptions of these functions.

[data,geom] = pre_recon(pathname, filename, geom_type, slices); >> Loading the SophiaBeads dataset (3D)... >> Dataset is cut down to 1564 x 1564 x 200... >> Applying centre of rotation correction... >> Pre-reconstruction stage is complete!

Reconstruction and Saving

We are now ready to reconstruct the SophiaBeads dataset. To encourage readers to implement their own techniques, we include a short CGLS script in the project codes release, named cgls_XTek. The readers are free to use this code to obtain the SophiaBeads dataset result in these tutorials, or as a template to implement own techniques. The CGLS algorithm is implemented as described in [2], and run for 12 iterations (this was picked by trial and error, as explained in [5]). The program command and the outputs are

xcgls = cgls_XTek(data, geom, iterations);

```
>> Reconstructing the SophiaBeads dataset (3D)...
>> Iteration 1 -- Elapsed time is ---- seconds.
>> ...
>> Reconstruction is complete!
```

The code then plots the reconstructed image (for 3D, this is just the central slice).



Figure 3: This is the central slice of the reconstructed volume.

The SOPHIABEADS_512_AVERAGED dataset is now reconstructed. The volume is saved as single floats, using the script write_vol.m:

```
volname = write_vol(xcgls, pathname, filename, experiment_name, 'single');
```

which outputs

```
>> The reconstructed volume is written in folder
    /media/SophiaBeads_Datasets/SophiaBeads_512_averaged/.
```

>> The volume is saved as SophiaBeads_512_averaged_CGLS_200slices.vol.

The reconstructed result for this dataset is now ready to be exported to Avizo for the quantification stage.

5.2 Quantification using Avizo

For this section of the tutorials, we have divided the content into three sections:

- I. Loading the Dataset,
- II. Image Segmentation and Separation, and
- III. Quantification and Saving Results.

I. Loading the Dataset

Start the Avizo Fire application and, from the top of the page, select $\texttt{File} \rightarrow \texttt{Open Data} \rightarrow \texttt{Navigate}$ to the folder where $\texttt{write_vol.m}$ has saved the dataset. The window in Figure 4(a) will appear. Select "*Read complete volume into memory*", and press OK. The reader will then be prompted to pick the file format. Pick "*Raw Data*" from the list (as highlighted in Figure 4(b)) and press OK.



(e) The setup of Avizo.

Figure 4: Data loading in Avizo.

The reader will then be asked to enter a set of dataset parameters (see Figure 4(c) for the default values, and Figure 4(d) for the changes applied for the SophiaBeads Datasets). Apply the changes below:

- Data type: Select 32-bit float from the drop-down menu.
- **Dimensions:** This should be the size of the reconstructed volume, which in this case is $1564 \times 1564 \times 200$.

• Voxel size: This should be as defined in the .xtekct files, which (for all SophiaBeads Datasets) is [16, 16, 16]µm (this can be left as [1, 1, 1] but in doing so, the results obtained during the analysis stage would have to be scaled).

Note that after these changes are applied, the **Header** should read 0 (zero). After loading the dataset, one can attach **Ortho Slices** to view the horizontal (default orientation) and vertical slices (choose xz and yz as the *Orientation* option in the **Properties** window below the **Project View**, on the left hand size of the application). The current setup should now match the one given in Figure 4(e).

II. Image Segmentation and Separation

The first step is to get the binary image (this is to separate the objects from the background). For this, we apply the action *Interactive Thresholding*: Right click on the dataset (.vol) \rightarrow Image Segmentation \rightarrow Binarization \rightarrow Interactive Thresholding \rightarrow Create. See Figure 5(a) as a reference. A red (action) box will appear below .vol with options to modify. The only modification needed here is the *Intensity Range*, which should be from 0.11 to 0.35 (in fact, anything above 0.35 will not effect the analysis results). This range must be the same for all the reconstructed results of SophiaBeads Datasets. Figures below are of the option window and an ortho slice attached to the output of this action.

Project View		5 ×	• • • • • • • •	Prope	rties	₽×
Ortho Slice FFT Filter	Volume Rendering Filter By Measure			0	Interactive Thresholdin	
SophiaBeads_512_average	CGLS 200slices.vol* 🔿 👞 👘	Ortho Sice ()		平	Data:	SophiaBeads_512_averaged_CGLS_200slices.vol 💌 🛋
	SophiaBeads_512_aGLS_200slices Favorites	i.vol + 👩 🔗 Q inter	Adantive Thresholding	至	Mask:	NO SOURCE 🔻 🔿
	Editors	20 2D-Histogram Segmentation	Hysteresis Thresholding Interactive Thresholding	臣	Preview Type:	✓ 2D 3D
	Templates	Correlation Histogram	O Interactive Top-Hat	平	Preview Orientation:	● xy ○ xz ○ yz
	Animate Annotate	 Hierarchical Watershed Multi-Thresholding 	Local Infestiolding	포	Colormap:	0 0.337359 Edit.
	Convert	Watershed Segmentation	l i	平	Preview Slice Number:	▶ 100
	Display Geometry Transforms			푸	Rendering:	⊖ transparent
	Image Processing			平	Intensity Range:	0.11 0.35
	Measure And Analyze			포	Color Mask:	

(a) Choosing Interactive Thresholding.



(c) Resulting ortho slice of the binary image (blue = objects of interest, black = exterior).

Figure 5: Interactive Thresholding steps for a binary image of the volume.

(b) Options.

Next, we apply the Chamfer Distance Map to the binary image. To do this, right click on .thresholded \rightarrow Image Processing \rightarrow Distance Maps \rightarrow Chamfer Distance Map \rightarrow Create (Figure 6(a)). Note that in the **Properties** window, the reader must choose 3D for the Interpretation option and select Apply. Attaching an ortho slice to the output gives Figure 6(b). This is followed by the H-Maxima action: Right click on the Distance Map output, .distmap \rightarrow Image Processing \rightarrow Morphological \rightarrow H-Maxima \rightarrow Create. No changes needed to make in the properties, so click Apply. It might take a few minutes to complete the iterations. Finally, we need to attach Labeling to the H-Maxima output. As before, right click on .hmaxima \rightarrow Image Segmentation \rightarrow Labeling \rightarrow Create and then Apply.

 SophiaBeads_512_averaged_CGLS_200slices.tl 	SophiaBeads_512_avslices.thresh	olded 🗸 📑 🚰 🔍 <enter a="" sea<="" th=""><th>arch string></th><th>0</th></enter>	arch string>	0
	Favorites Favorites Favorites Favorites Animate Animate Compute Convert Display Geometry Transforms Image Processing Image Segmentation Measure And Analyze	Distance Maps Gge Detection Fequres Selection Frequency Domain Grayscale Transforms Propagation Propagation Separating And Filling Sharpening Skeletonization Smoothing And Denoising Filter Sandbox	Chamfer Distance Map Chessboard Distance Map Closest Boundary Points Distance Map Distance Map Distance Map For Skeleton Geodesic Distance Map	, C

(a) Choosing Chamfer Distance Map.



(b) Distance Map Output.

512_averaged_CGLS_200slices.distmap	SophiaBeads_512_a200slices.dist	map 🗸 📝 🖻 🔍 <enter a="" sear<="" th=""><th>ch string></th></enter>	ch string>
	Favorites Favorites Favorites Favorites Favorites Favorites Favorites Annotate Compute Convert Display Geometry Transforms Image Processing Image Processing Messure And Analyze	Distance Maps Didge Detection Frequency Domain Grayscale Transforms Morphological Operations Propagation Separating And Filling Sharpening Skeletonization Smoothing And Denoising Filter Sandbox	Closing Closing Clistion Frosion Grayscale Reconstruction H-Maxima Maxima Imposition Minima Imposition Opening Regional Maxima Regional Minima

(c) H-Maxima.

SophiaBeads 512 averaged CGLS_200slice	s hMaxima* ①		
	SophiaBeads_512_a200slices.hMa	axima 🗸 📝 🕋 🔍 <enter a="" sea<="" th=""><th>rch string></th></enter>	rch string>
	Favorites Recents Editors	Binarization Add To Label Connected Components	<mark>ې Labeling</mark> 🖄 Image labeling
	Animate Annotate Compute Convert Disclary	Hierarchical Watershed Interpolate Labels Labeling Multi-Thresholding Remove Small Holes	Type: label Former Name: label
perties SophiaBeads_512_averaged_CGLS_200slice	Geometry Transforms Image Processing Image Segmentation	Semove Small Spots	
Lattice info: 1564 x 1564 x 200, unifo	Measure And Analyze		

(d) Labeling.

Figure 6: Steps for detecting individual objects.

Now, we must go back to the distance map and invert it: Right click .distmap \rightarrow Compute \rightarrow Logical Operations \rightarrow NOT \rightarrow Create. Nothing to change in the **Properties** window so click Apply. We are now ready to apply the Watershed algorithm on the inverted image: Right click on the output .not \rightarrow Image Segmentation \rightarrow Marker-Based Watershed \rightarrow Create. On the **Properties** window, for the *Input Label Image*, select the .labels output. See Figure 7(c) for the **Properties** window changes, and Figures 7(d) and 7(e) for the results of .not and .watershed.

SophiaReads 512 averaged CGLS 200sices distman			
SophiaBeads_512_a200sices.dis	stmap 🗸 📑 🚰 🍳 <enter a="" se<="" th=""><th>arch string></th><th>0</th></enter>	arch string>	0
DirkMaanna O SophisBeads_512_averaged_CGL5_200skces.labels* SophisBeads_512_averaged_CGL5_200skces.labels* Anmotate Compute C	Arithmetic Operations Logical Operations Volume Operations Arithmetic Calculus MATLAB Compute Tensor Create Automatic Landmarks DICOM Send Filter By Spreadsheet Gradient Interpolate	AND Image AND NOT Image AND NOT Image AND NOT Value AND Value Bit Shift Invert NAND Value NAND Value NOT NOT NOC Value NOT NOC Value NOT NOC Value NOT NOC Mage NOCR Mage NOCR Value	Perf

(a) Choosing the NOT option.

SophiaBeads_512_averaged_CGLS_200slices.no	SophiaBeads_512_aGLS_200slice	s.not 🗸 📑 🚰 🔍 <enter a="" sea<="" th=""><th>rch string></th></enter>	rch string>
	Forontese Forontese Forontese Editors Forontese Annotate Compute Compute Compute Compute Geometry Transforms Image Processing Image Segmentation Messure And Analyze	Binarizion Binarizion Components Correlation Histogram Correlation Histogram Edit New Label Field Marker Based Watershed Multi-Thresholding Watershed Segmentation	Marker-Based Watershed Performs a fast determination of the watershell Type: catchbaan Former Name: catchbaan/fastwa Oreate More Info

(b) Choosing the Watershed algorithm.

Prope	rties		đΧ
0	Marker-Based Watershed		8 ?
프	Input Grayscale Image:	SophiaBeads_512_averaged_CGLS_200slices.not	4
포	Input Label Image:	SophiaBeads_512_averaged_CGLS_200slices.labels 💌	
至	Туре:	Watershed 👻	
至	Interpretation:	3D O XY planes	
至	Neighborhood:	○ 6 ○ 18	
至	Algorithm Mode:	fast 💌	
	auto-refresh		Apply

(c) Watershed options: Choose Input Label Image.



Figure 7: Steps for the Watershed algorithm application.

Finally, subtract the separation lines (.watershed) from the labeled binary image (.thresholded) by right-clicking on .thresholded \rightarrow Compute \rightarrow Logical Operations \rightarrow AND NOT Image \rightarrow Create. On the **Properties** window, choose .watershed output from the drop-down menu for the *Input Image 2* (see Figure 8(b)). Clicking Apply produces the output .sub. Repeat the steps for the *Labeling* process to index the separated objects. We can now quantify these objects (reconstructions of the individual beads) using the measures listed in §4.

SophiaBeads_512_averaged_CGLS_200slices.watershed	SophiaBeads_512_a0slices.waters	shed 🗸 📑 🚰 🔍 <enter a="" sea<="" th=""><th>rch string></th><th>0</th></enter>	rch string>	0
opertiles	Sophadeda <u>512</u> <u>a</u> .usides.waters Favorites Recents Animate Annotate Compute Convert Display Geometry Transforms Image Processing Image Processing	Arithmetics Operations Logical Operations Volume Operations Volume Operations Arithmetic Calculus MATLAB Create Automatic Landmarks Create Label Colormap DICOM Send Filter By Spreadsheet Generate Lego Surface Generate Surface	AND Image AND NOT Image AND NOT Image AND NOT Value Bit Shift Invert NAND Value NAND Value NOR Image NOR Value NOR Value NOR Value NOR Value NOR Value	Perform Type: 1 Forme
SophiaBeads_512_averaged_CGLS_200slices.watershe	Measure And Analyze	 Interpolate Relabel 	NXOR Value OR Image OR V	~

(a) Choosing the AND NOT Image.

Properties		₽×
AND NOT	Image	8
T Input Ima	ge 1: SophiaBeads_512_averaged_CGLS_200slices.thresholded ▼ 🖨	
T Input Ima	ge 2: SophiaBeads_512_averaged_CGLS_200slices.watershed	
	NO SOURCE SophiaBeads_512_averaged_CGLS_200slices.vol SophiaBeads_512_averaged_CGLS_200slices.thresholded	
	SophiaBeads_512_averaged_CGLS_200slices.distmap SophiaBeads_512_averaged_CGLS_200slices.hMaxima	
	SophiaBeads 512_averaged_CGLS_200slices.labels SophiaBeads_512_averaged_CGLS_200slices.labels	

(b) AND NOT Image option.





(c) Visualization using *Surface View* of *Labeling* 2 (vertical view).

(d) Visualization using *Surface View* of *Labeling* 2 (tilted view).



(e) Labeling 2; Separated objects.

Figure 8: Final steps for separating the objects.

III. Quantification and Saving Results

We now have our reconstructed beads separated and individually labeled. The second label lists that there are 4329 objects (this is visible in Figure 9(a)). However, this does not mean that there are exactly 4329 beads: this number takes into account any objects of any size within the image. These include any beads that are not fully in the 200 slices window, which may or may not have been separated from other beads (see Figures 8(c) and 8(d) for the *Surface View* of the reconstructed volume). To quantify our reconstruction appropriately, we need to filter out the objects that are too small to be included in the analysis. This is done by *Filter by Measure*: Right click on the second .labels \rightarrow Measure and Analyze \rightarrow Individual Measures \rightarrow Filter by Measure \rightarrow Create. We believe the most appropriate measure for filtering out is the VOLUME3D option, which sorts the objects by the descending order of their volume. See Figure 9(b) for the modified properties. Note that we pick the first 50 objects that have the highest volume. These are saved in the *third* output of .labels.

 SophiaBeads_512_averaged_CGLS_200slices2. 	SophiaBeads_512_av200slces2.	labels 🗸 📑 🚰 🔍 <enter a="" se<="" th=""><th>arch string></th><th>Properties</th><th></th><th>ð×</th></enter>	arch string>	Properties		ð×
112_mvmaged_CGL5_20054ces2.labels	Favorites Favorites Forcents Generals Annotate Compute Compute Convet Diplay Geometry Transforms Image Processing Image Processing Measure And Analyze	Global Messures Individual Messures Hidtogram Hidtogram Inage Curvature IL Line Probe Proint Probe Proint Probe Prostiles Analysis Wizard Spine Probe Volume Fraction	Filter By Measure Label Analysis Sieve Analysis Filter Serve Analysis Filte	Import Image: Soot Import Image: Soot Import Interpretation: ⊕ 1 Image: Heasure: Heasure: House Heasure: House Image: Heasure: Heasure: House Image: Soot Image: House Image: Soot Image: Soot Image: Soot Image: Soot	hisBesds_512_everaged_CGL5_2008iccs2.labels hisBesds_512_everaged_CGL5_2008iccs.vol 20 O XY planes ma3d • hest Values •	 Apply

(a) Choosing Filter by Measure.

(b) **Properties** window.



(c) Volumized Rendering of the resulting .labels.

Figure 9: Filtering out the objects that are small in volume.

The next stage involves some manual work. The third output of .labels contains 50 objects and these are sorted by the magnitude of their volumes. What we are not taking into account here is that some objects are not separated properly, or are *mostly* within the 200 slices range so they are able to go through the filter (as seen in Figure 9(c)). We need to pick out the beads within these 50 objects that are separated and are fully within the window. To do this, we have to use the **Segmentation Editor** where all objects are listed as individual materials. In Figure 10(a), we see the first 5 objects (viewed by toggling the 3D option). From this we deduce that Material 3 is not separated properly, and Materials 1, 2 and 4 are not fully in the window. For the quantification, we can only accept Material 5 so we delete the first four materials off the list using the *Delete* key (they are simply added to the exterior and are no longer part of the third .labels output). Using this method, we inspect all the materials on the list. Our target is to keep around 10 materials for the *Label Analysis* stage (this can be challenging as the quality of reconstructed volume decreases, in which case we would go back to *Filter by Measure*, and increase the *Number Of Objects* option). See Figure 10(c) for another example, and Figure 10(d) for the resulting 10 materials we accept for the next stage. Please note that the colours may be different but by following the exact steps, the reader should be able to get the beads in the same locations.



(a) The first 5 materials on the list: Materials 1 to 4 are not suitable.

A										
File	Edit	Project	View	Window	XPand	XScreen	XTeam	He	lp	
A	START	r	Ŕ	PROJECT		SEC SEC	MENTATIO	N	22	ANIMA

(b) Segmentation Editor window.



(c) A second example of beads: Materials 5 and 9 are acceptable for the analysis.



(d) Final look of the appropriate beads in the Segmentation Editor.

Figure 10: Manually choosing beads that are separated and fully within the 200 slices.

The final stage is running the *Label Analysis*, which concludes the quantification. To perform the *Label Analysis*, right click on the third .labels output \rightarrow Measure and Analyze \rightarrow

Individual Measures \rightarrow Label Analysis \rightarrow Create (Figure 11(a)). In the Properties window, choose .vol for the *Intensity Image*. For convenience, we created a measure group for the SophiaBeads experiments. This is done by clicking ... next to *Measures*. This prompts the window shown in Figure 11(c). To create a new measure group, press on top of the page, next to the "*Choose a measure group option*", and enter a name for a new group. Then, browse the "*Native measures*" section and double click on the preferred measures to include in the group. The measures picked should appear on the right, under the section "*Measures selected in the group*". Press OK when finished, and make sure the newly created measure group is chosen as the *Measures* option in the **Properties** window (compare with Figure 11(b)). Clicking Apply will prompt Avizo to measure the remaining objects in .labels, and output results in a table for all the measures in the group. The resulting table for our run is given in Figure 11(d). By inspecting these results, we can quantify how close the reconstructed volume is to a perfect solution, and thus compare with different methods applied to the same volume (see e.g. [6, Fig. 7]).

 DB SophiaBeads_512, 	sveraged_CGLS_200slices3.labels* ① SophiaBeads_512_av200slices3.	labels 🗸 📑 🚰 🔍 <enter a="" sea<="" th=""><th>arch string></th><th>0)</th></enter>	arch string>	0)
512_averaged_CGL5_200 1564 x 1564 x 200, u label, 8-bit unsigned,	favorites forecrites forecrites favorites favorites favorites favorite favorite favorite foropute Convet Geometry Tandforms mage Processing mage Segmentation Measure And Analyze	Global Messures Global Messures Global Messures Histogram Glupage Curvature Hat Line Probe Hat Line Sch Probe Glupage Material Statistics Hat Pinit Probe Prostilise Analysis Wizard Glupage Probe Volume Fraction	 Filter By Measure Label Analysis Sieve Analysis 	Cor Typ. For

(a) Choosing Label Analysis.



(b) The final **Properties** window of *Label Analysis*.

٥	Selection o	f mea	sure groups	×		•			Table	s		-	□ ×
Choose a measure group: So	phiaBeads_Measures 👻		- I I		F	SophiaBea	ads_512_avera	ged_CGLS_200	slices3.Label-A	nalysis 🗗 🗙			
User measures:			Measures selected in the group:				1	<u>ull</u> 🕀					
Name	Formula		Name Volume3d EqDiameter Area3d Shape_VA3d Symmetry Excentricity	Formula Native Native Native Native Native Native		Mean Min Max Median Variance Kurtosis	voiume3d 8.79831e+09 7.40989e+09 1.0982e+10 8.83652e+09 1.14513e+18	EqDiameter 2557.25 2418.83 2757.79 2564.84 10414.7 -0.602641	Area3d 2.44956e+07 1.93694e+07 2.88112e+07 2.47988e+07 9.04389e+12 -0.968621	Shape_VA3d 1.70264 1.10405 2.12112 1.75334 0.120246 -1.04148	index 5.5 1 10 5 8.25 -1.22425	Materials	^
		0				Skewness	0.593039	0.468217	-0.274873	-0.516507	0		~
Native measures:		<					Volume3d	EqDiameter	Area3d	Shape_VA3d	index	Materials	^
Name	Formula ^					1	1.0982e+10	2757.79	2.88112e+07	1./5334	1	Material5	
Anisotropy	Native					2	1.00455e+10	2677.06	2.83198e+07	1.99011	2	Materia	
Area	Native					3	9.29738e+09	2608.88	2.39781e+07	1.41017	3	Materia	
AreaSc	Native					4	9.2375e+09	2603.27	2.55478e+07	1.72781	4	Materia	
BaryCenterX	Native					5	8.9129e+09	2572.41	2.67107e+07	2.12112	5	Materia	
BaryCenterY	Native					6	8.47617e+09	2529.69	2.57804e+07	2,10872	6	Materia	
BaryCenterZ BinMom2x	Native					7	8.0601e±09	2487.6	2 00028e+07	1 10/05	7	Materia	
BinMom2y	Native						7.00252 00	2470.24	2.005200+07	1.710503		Marchida	
BinMom2z	Native					8	7.89253e+09	2470.24	2.29558e+07	1./1003	8	Materia	
BinMomxy	Native 🗸					9	7.66908e+09	2446.7	2.33918e+07	1.92421	9	Materia	
			OK	Cancel Help		10	7.40989e+09	2418.83	1.93694e+07	1.17024	10	Materia	

(c) Creating measure groups in Avizo.

(d) The final table with label analysis results listed.

Figure 11: Final steps of quantifying the quality of a reconstructed volume using Avizo.

It is possible to visualize the results in 3D by applying *Volumized Rendering* to the final .labels output, or by applying *Generate Surface* to the same output, followed by *Surface View* (the outputs will be similar to Figures 9(c) or 8(d)). The tree flow in the **Project**

View window should match the one given in Figure 2. The reader can save the results on the analysis table in a csv or xml format.

6 Additional Notes and Licensing Information

Due to the nature of the sample, the datasets are also suitable for developing and/or testing

- segmentation methods,
- image or data correction techniques,
- forward and back project implementations, and
- benchmarking own codes or method.

We include mex files suitable for Windows, Mac OSX and Linux operating systems. We also include the Linux template of the script we have written to create these mex files (setup.m). Please note that as this is a template script, it may not work with Windows, and may need some tweaking to work for Mac OSX systems (this depends on the installed versions of MAT-LAB and Xcode).

If the reader runs **setup.m**, the existing projector and back projector mex files will be overwritten. To avoid this, modify the names after the **-output** option in the **setup.m**:

Line 22: -output mex/CBbackproject_c

Line 23: -output mex/CBproject_c

Please note that in doing so, the reader will also have to modify the interface scripts for the mex files (CBproject.m and CBbackproject.m) to update the lines with the new projector and back projector mex file names.

Finally, the project codes are tested using the following MATLAB versions:

- Scientific Linux 6, Ubuntu 12.04 and 14.04: MATLAB R2010a, R2013a, 2014b.
- Mac OSX 10.9 (Mavericks) and 10.10 (Yosemite): MATLAB R2009b, R2013a, R2014b.
- Windows 7 and 8.1: MATLAB R2012b, R2013a, R2014a, R2014b.

Licensing Information

The SophiaBeads Datasets are distributed under the CREATIVE COMMONS ATTRIBUTION SHARE-ALIKE license. The project codes are distributed under the GPL v2 and MIT licenses. Please read the documents LICENSE and LICENSE2 to learn more about using the project codes. The reader may also find it useful to visit www.creativecommons.org/licenses/ and www.choosealicense.com to learn about these licenses.

Acknowledgments

This project is funded by the School of Mathematics, EPSRC CCPi (EP/J010456/1) and BP through the ICAM framework.

The author is grateful to the MXIF staff Dr. Sam McDonald and Dr. Julia Behnsen for their help with acquiring the datasets and with visualizing the results. The author is also grateful to the PhD supervisors Prof. Bill Lionheart and Prof. Phil Withers for their helpful comments and guidance.

References

- [1] Avizo 8. Avizo User's Guide. English. FEI Visualization Sciences Group.
- [2] Å. Björck, T. Elfving, and Z. Strakos. "Stability of conjugate gradient and Lanczos methods for linear least squares problems". In: SIAM J. Matrix Anal. Appl. 19(3) (1998), pp. 720–736.
- S.B. Coban. SophiaBeads Datasets Project Codes. Apr. 2015. DOI: 10.5281/zenodo.
 16539. URL: http://dx.doi.org/10.5281/zenodo.16539.
- S.B. Coban and S.A. McDonald. SophiaBeads Datasets. Mar. 2015. DOI: 10.5281/ zenodo.16474. URL: http://dx.doi.org/10.5281/zenodo.16474.
- [5] S.B. Coban et al. "A framework for exploring the limits of reconstruction algorithms". In: *(to appear)* (2015).
- [6] S.B. Coban et al. "When do iterative reconstruction methods become worth the effort?" In: the proceedings of The 13th International Meeting of Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine. 2015.
- T. Liu. "Direct central ray determination in computed microtomography". In: Optical Engineering 48.4 (2009), pp. 046501–046501.
- [8] Manchester X-Ray Imaging Facility Nikon Custom Bay. http://www.mxif.manchester. ac.uk/resources/imaging-systems/nikon-custom-bay. Accessed: 20-MAR-2015.
- [9] SophiaBeads Datasets Project Codes GitHub Page. http://sophilyplum.github. io/sophiabeads-datasets/. Accessed: 01-APR-2015.
- [10] E. Sundermann et al. "A fast algorithm to calculate the exact radiological path through a pixel or voxel space". In: (1998).
- [11] D. Szotten. "Limited Data Problems in X-ray and Polarized Light Tomography". PhD thesis. School of Mathematics, University of Manchester, 2011.
- [12] W.M. Thompson. "Source firing patterns and reconstruction algorithms for a switched source, offset detector CT machine". PhD thesis. School of Mathematics, University of Manchester, 2011.
- [13] N. Wadeson. "Aluminium foam data reconstruction using CGLS and TV Regularisation - 100 and 200 projection data." MIMS Preprint. 2012.